Digital Phase Follower -- Deserializer in Low-Cost FPGA

Jinyuan Wu, Z. Shi

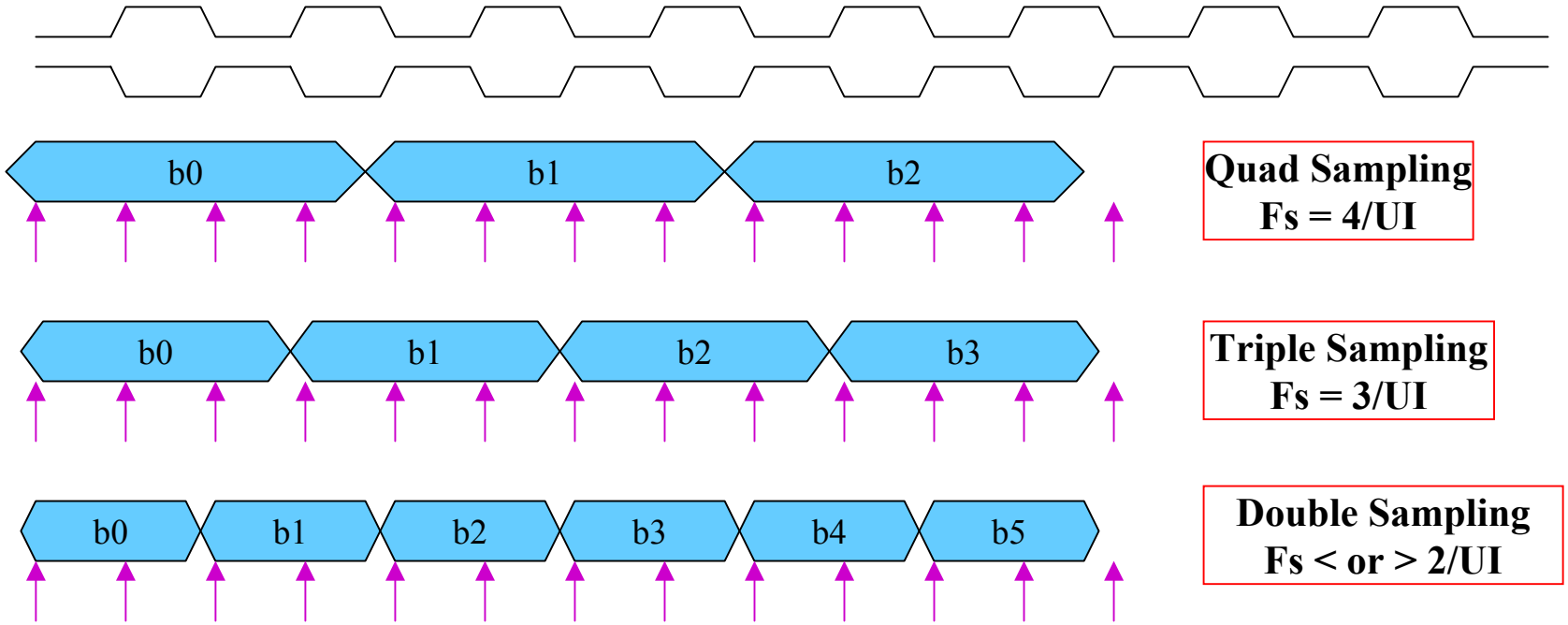
Motivation

- In HEP systems, sometime many channels of serial data must be concentrated.
- It will be nice if the data clock is not transmitted separately. (Just transmit a single data channel).
- It will be nice if it can be received in low-cost FPGA in which dedicated serial data receivers are not available.
- It will be nice if user protocol can be supported. (Can be 8B/10B, or can be anything users want).
- Examples:
  - TSO modules to PP modules. (500 Mbps, user protocol).
  - FPIX2 to PDCB. (140 Mbps, user protocol).

Receiving Serial Data

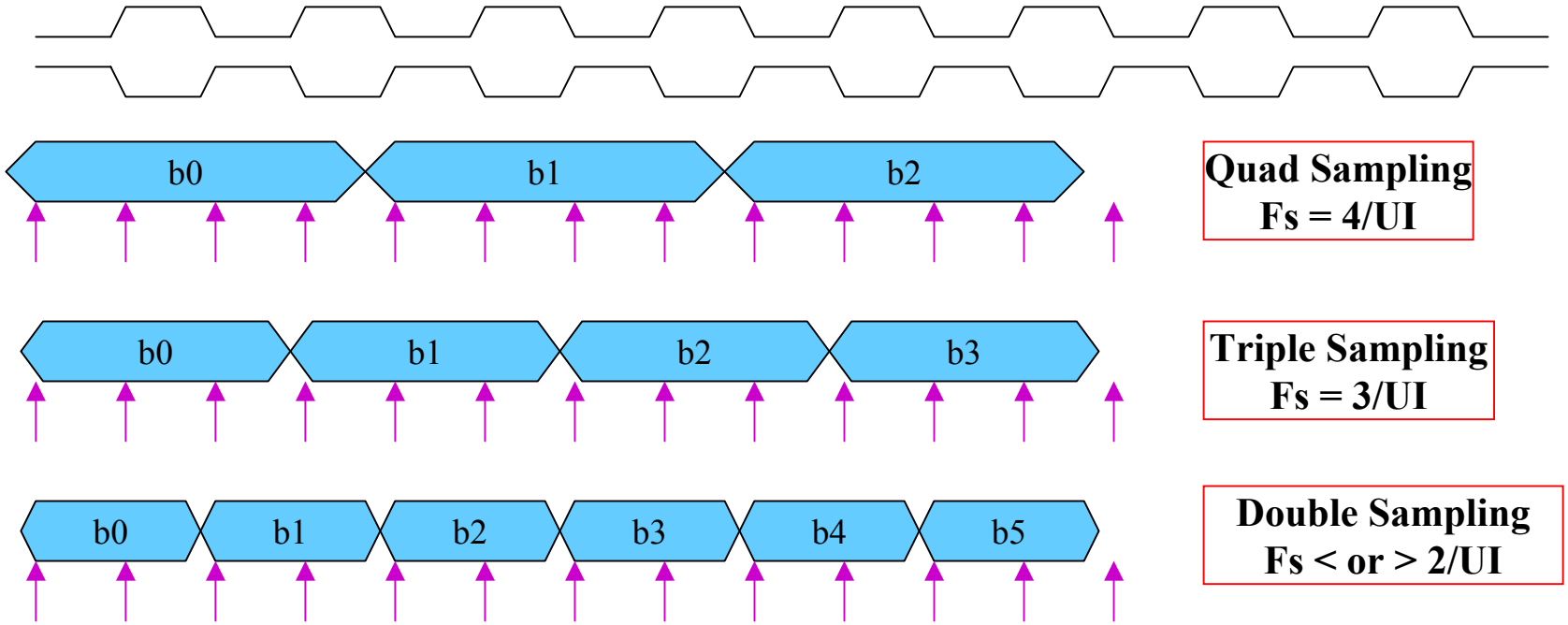
- Data channels are de-serialized using shift registers.
- The clock for the receiving shift registers comes from:
  1. Separate channel. (Channel-channel skew ☹).
  2. Same data channel.
     - Clock recovery using PLL. (Phase detection+VCO).
     - Dynamic phase aligner. (In Altera devices, choosing a correct clock phase from 8 phase samples).
     - Digital phase follower. (For low-cost FPGA).

Multiple Sampling



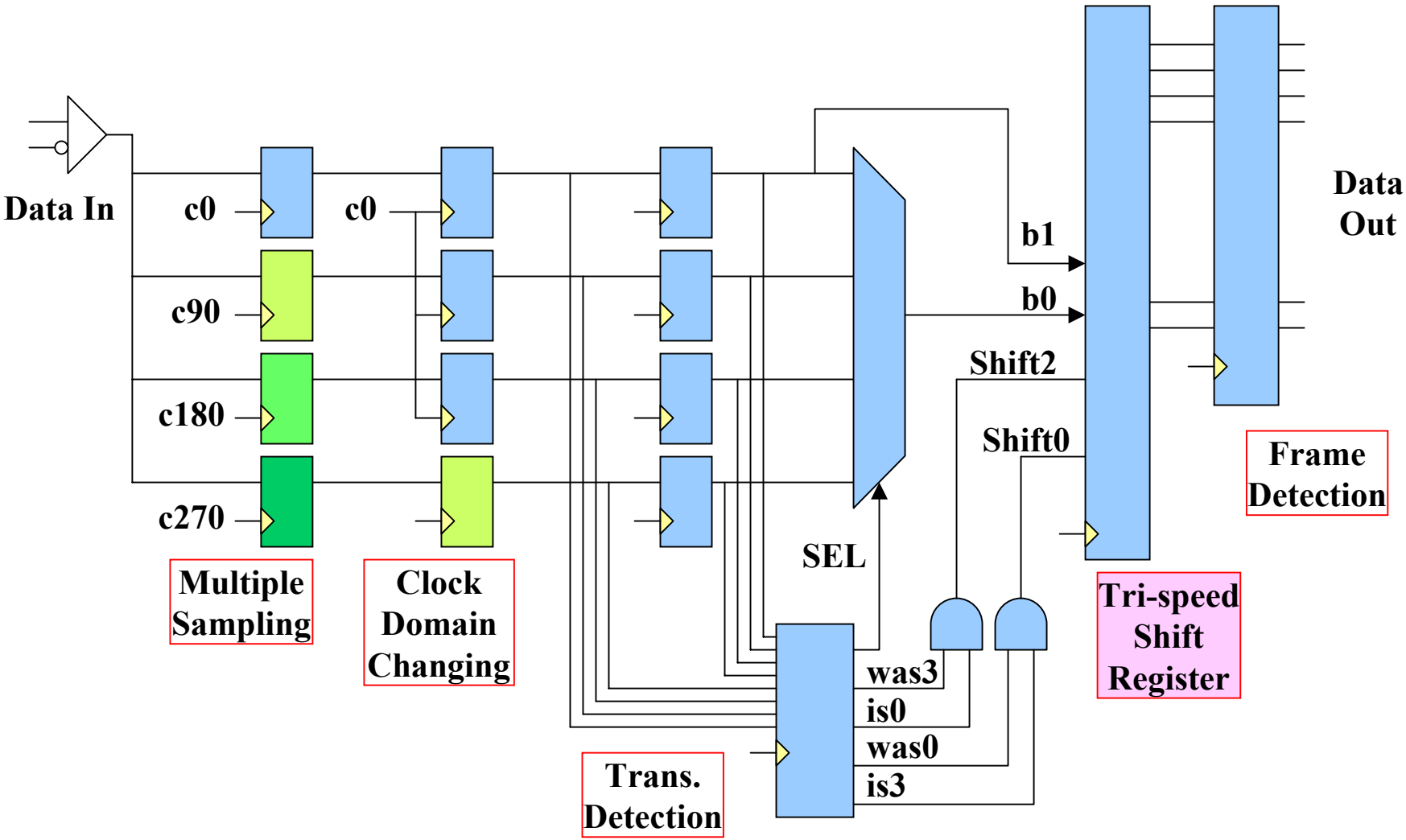
- Multiple sampling is used to determine the phase of the data.
- A correct sampling point is automatically chosen after first 0 to 1 transaction.
- The sampling point shifts following the shift of the data phase.
- Everything is in standard digital circuit.

More Notes on Multiple Sampling

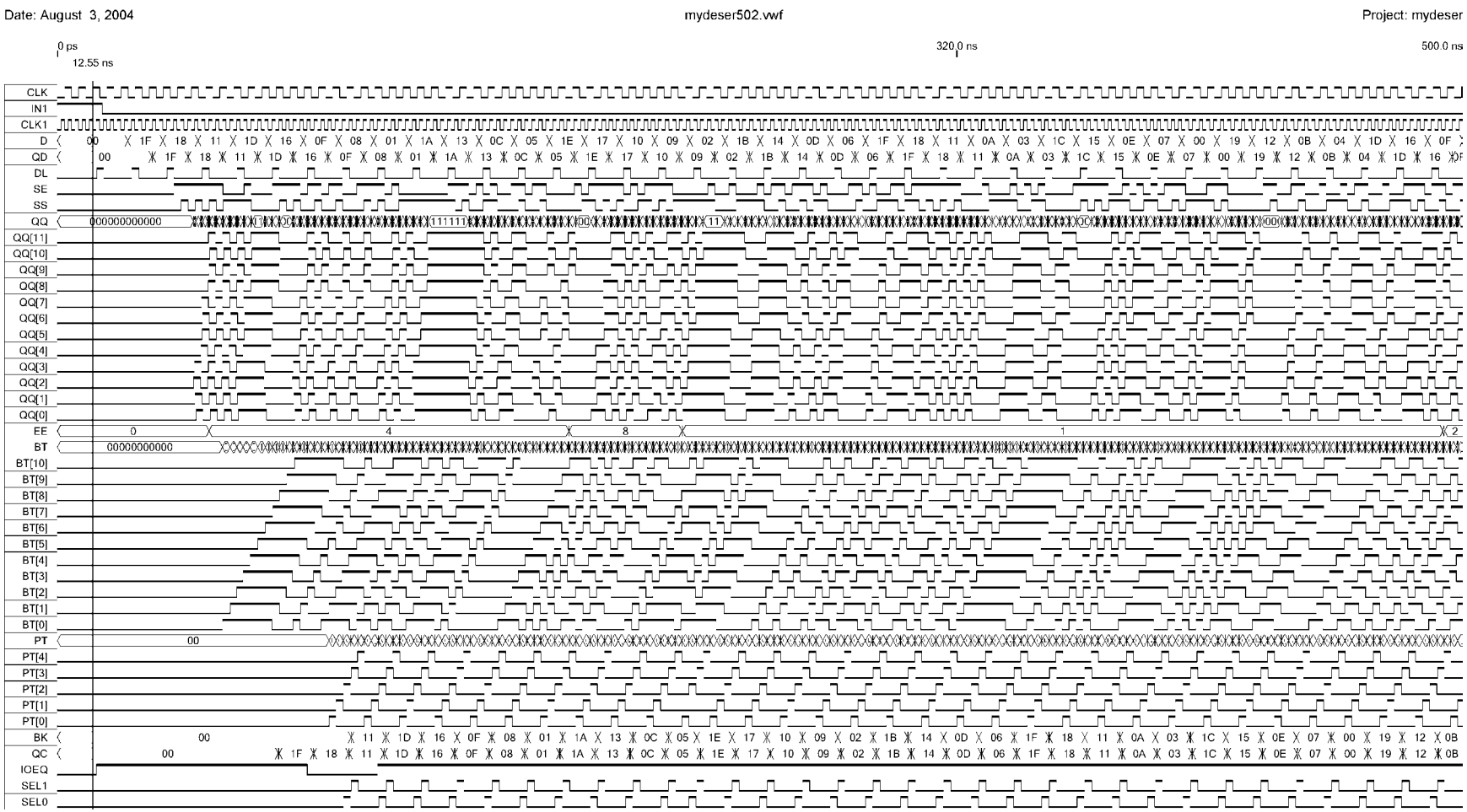


- In digital phase follower, since no clock recovery is needed, 4, 3 or 2 samples per bit (unit interval) are sufficient. (Not 8).
- In double sampling case, sampling rate must be known either less or larger than 2/UI.

Digital Phase Follower

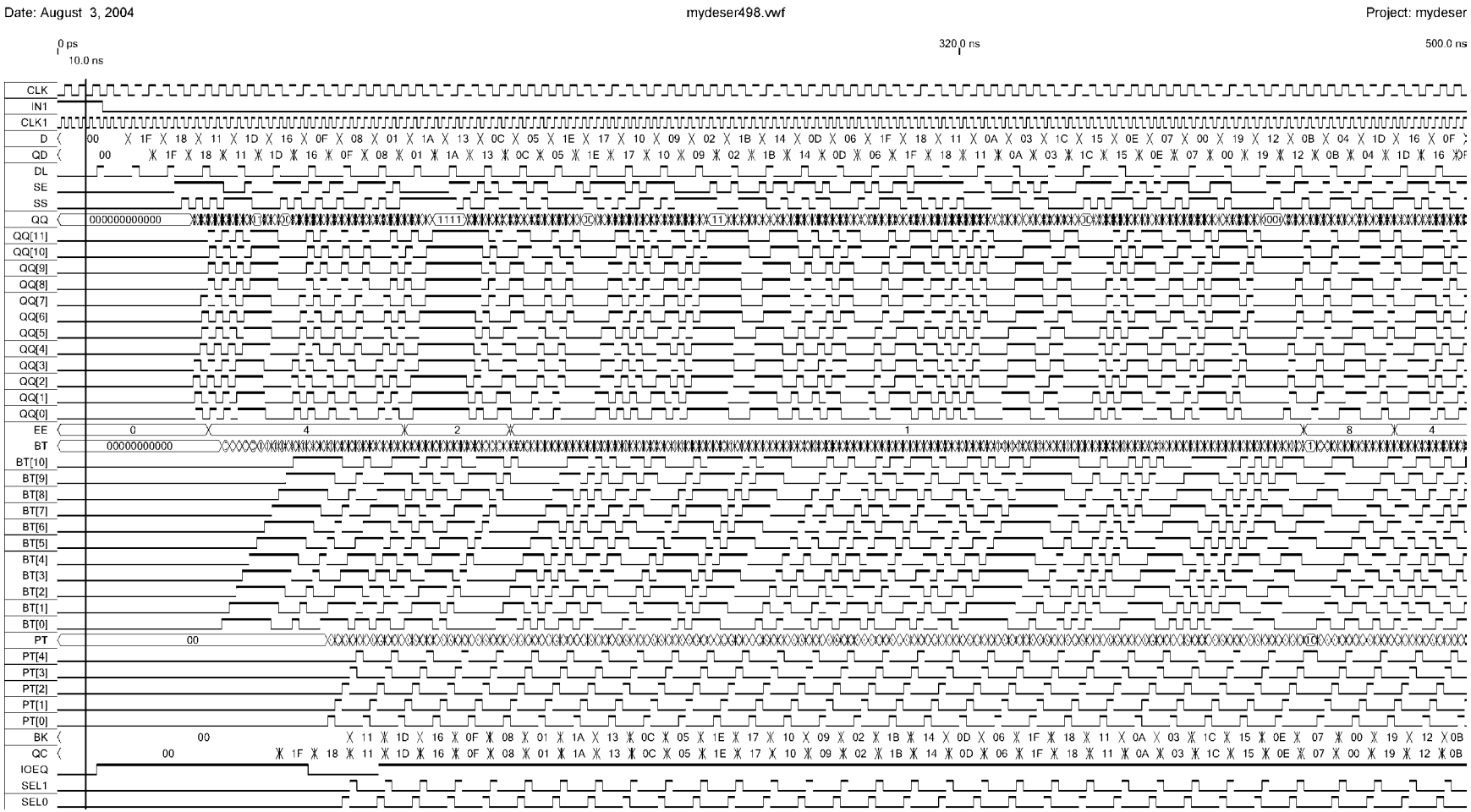


Simulation (1)



- This is a 4B/5B receiver working at 400Mbps, compiled in an Altera Cyclone device.
- The receiving clock is 0.4% slower – no errors is seen.

Simulation (2)



- The same receiver running with receiving clock 0.4% faster – no errors is seen.

Deserializer Based on Digital Phase Follower

- Data is self-timed, no separate clock transmission is needed.
- The transmitter and receiver clocks can be independent – frequency difference is compensated.
- User protocols are supported.
- It can be implemented in low-cost FPGA.